

IN THE SPECIFICATION:

Kindly amend the specification at pages 3-5, 17-22, 24, 29, 33-36, and 43 as follows.

Page 3, line 7 through page 4, line 2:

The opportunity tracking information system of the present invention includes a method for managing tasks in a computer communication network. The method includes accessing a centralized database[[,]] containing information on tasks and their status. The status of each task includes, but is not limited to: a description of the task, and the person (or persons) within the organization responsible for completing the task. The program used to access the server is preferably a browser or other computer program designed for access across a network. After identifying the person using the program as the “user,”[[,]] the status of all tasks associated with that user or individuals for whom the user has authority to monitor is displayed in a formatted manner. The user may then update the status of tasks that he is permitted to view. In general, a user may view the status of a task if the user is responsible for accomplishing the task or if the person responsible for accomplishing the task reports to the user within the organizational structure. Thus, the ability to view tasks parallels an organizational chart since a user at a position in the organizational chart can view tasks for that position and tasks for any position in the organizational chart below that position. Among the updates that the user may accomplish are creating new tasks for himself or for those who report to him, marking tasks as being finished, and assigning (moving) a task to other individuals or positions within the organization.

Page 4, line 18 through page 5, line 15:

In one key aspect of the present invention, the opportunity tracking information system merges the inherent organizational structure of an organization ~~inherently~~ into the processes for creating, editing, moving, adding, removing or viewing tasks. In particular, the opportunity tracking information system provides three general types of users: users, managers and

administrators. A user refers to a position in the organization chart that does not have any nodes or positions below it. Thus, the system provides a user typically with only the ability to view tasks for a particular position or person; and can create, edit, add or remove tasks only for that position. A manager refers to a position in the organization chart that has at least one node or position below it. A manager has the same rights as a user but can also perform any of the actions on tasks for the nodes below (or reporting to) the manager position. The system therefore provides managers with the ability to view tasks for their position and any position below or reporting to their position; and the ability to create, edit, add or remove tasks their position and any position below or reporting to their position. An administrator refers to a position that is responsible for ensuring the system and the processes for tasks that the system allows match the organization chart of an organization. The administrator has the same rights as a user but can also modify control parameters for operation of the system. The administrators have the capability to change[[,]] which positions report to other positions, and thus, the tasks that each position may view, edit, modify or assign. It should be understood that a particular person could be both a manager and an administrator type and thus the system would allow that user to perform actions of either type. Accordingly [[ever]]every user will be able to perform different actions and view different data depending on the type of user they are.

Page 17, line 15 through page 18, line 5:

Referring now to Figure 30, the memory unit 1304 is shown in more detail. In particular, the portions of the memory 1304 needed for the processes of the present invention are shown and will now be described more specifically. As shown in Figure 30, the memory unit 1304 preferably comprises an operating system 1402, other applications 1404, at least one OTIS application 1408, a first module 1414, a second module 1418, a third module 1410 an a internet browser 1330. As noted above, the memory unit 1304 stores instructions and/or data that may be

executed by processing unit 1302. The instructions and/or data may comprise code for performing any and/or all of the techniques described herein. These modules 1402-~~[[418]]~~1418 are coupled by bus 1301 to the processing unit 1302 for communication and cooperation to provide the functionality of the system 100. Those skilled in the art will recognize that while the present invention will now be described as modules or portions of the memory unit 1304 of a computer system, the modules or portions may also be stored in other media such as permanent data storage and may be distributed across a network having a plurality of different computers such as in a client/server environment.

Page 19, line 18 through page 20, line 11:

Under the control of the first module, the server 103 communicates with the client computer 102. For example, the first module may be an IIS, which is the software module component of ActiveX that operates in a runtime environment enabling Active Server Pages (ASPs) to interface therewith. ASPs generally provide a framework for constructing Web applications using HTML, scripts and ActiveX components. The ASP page is created by embedding such scripts within the HTML page. As a user makes the request for an ASP page, the server 103 executes the scripts that have been embedded within the page so that the output generated from ~~[[the]]~~ running the scripts is included in the HTML, thus allowing a client browser on client 102 to view the page. It is noted that the present invention is well suited to work with other formats for creating forms and processing input, including Dynamic HTML (DHTML) technology. It will be evident to those skilled in the art that the client 102 is adapted to run various types of commercially available browsers (e.g., Netscape, Internet Explorer) suited to enable DHTML functionality. Furthermore, here and throughout this application, the description of the present invention in the context of the Internet, browsers, ASP, ~~[[[etc]]]~~etc. is only by way of example. Those skilled in the art will realize that the present invention could be

implemented on a variety of other hardware environments such as peer-to-peer networks and mainframe systems just by way of example.

Page 21, line 1 through page 22, line 5:

The present invention preferably uses COM objects to enforce rules about transactions. For example, if a business rule is that you want no addresses in your database that do not have valid zip codes then calling a COM object to validate the address may be used. In this case it might check that the zip code was valid for the city and state specified. Using a COM object allows many applications to share the same code. A COM object might be used to validate the due date for a ~~task e.g. to~~task, e.g., to verify that it is in the future and that it falls on a workday rather than a weekend or a holiday. The COM object might also check that the due date does not fall on scheduled vacation day for the user to whom it is assigned. The present invention includes a variety of COM objects to implement the functionality described below with reference to the process flowcharts and screen shots of FIGS. 2-28. The versatility of this approach lies in the coding of COM components, and the referencing of the COM components from within the ASP pages. Such COM object can reside on the server 103 used to serve up the objects. In the preferred embodiment, the second module 1418 or middle tier layer of OTIS 100 is used to enforce a variety of business rules. For example, the second module 1418 enforces access rights and viewing privileges. For users, the second module 1418 or middle tier restricts the display of tasks to only those tasks assigned to a particular user. For managers, the second module 1418 or middle tier restricts the display and performance of other operations on tasks to the tasks for the manager and those users assigned to report to the manager. Finally, the second module 1418 or middle tier ~~allow~~allows an administrator to establish roles, disciplines and projects to which a user or position is assigned. When a task is assigned to a position within the organization middle tier objects assure that only individuals assigned to that position can access it. A final example

of middle tier object functionality within OTIS 100 is the use of COM objects to report exceptions within OTIS 100. This reporting may be as simple as changing the status of a task that is past due or more complex such as sending an email at the time a task falls more than a predetermined time behind schedule. Again, while the present invention is described in the context of COM objects, it is only by way of example. Those skilled in the art will realize that the present invention could be implemented using other software constructs, code and methods.

Page 24, lines 5-22:

Reference is made to the flowchart of FIG. 3 to describe the process of determining whether a user has previously established a customized a display format or home page once the user has successfully logged into OTIS 100. A home page is the user's view of the data (i.e., the status information, which will be described subsequently in detail) in the system 100. Server 103 invokes a process to obtain 301 a home page for the user. To make this determination 302, server 103 accesses the database 104 to inquire about and retrieve the customized home page associated with the state values (e.g., cookie values) received in the request of for information (HTTP commands). If the user has not previously customized a home page (e.g., NO branch of 302), then a default home page is retrieved 303. The default home page is essentially a template with which status information can be incorporated. Next in step 304, using the state values (e.g., cookie values) that identify the user, the server 103 queries the database 104 to locate and retrieve 305 previously saved status information (e.g., tasks assigned, anomalies). The status information is then incorporated with the default home page (e.g., in DHTML form) and transmitted from server 103 for display 306 on client 102. For example, FIG. 7 illustrates one embodiment of a user interface having a default home page 700 combined with the status information 702 retrieved from database 104. If status information was not found on the database 104 for the particular user, then the default home page is displayed 306 without data.

Page 29, lines 3-23:

The functionality described above is integrated into a system 100 that automatically uses position and class to determine what operations a user is able to perform. An exemplary method for operation of the invention is shown in FIG. 32. The process begins by receiving 3202 a request for access to the system 100, and authentication 3204 of the user. Then based on the input user information, the position of the user is determined 3206. The position of the user determines whether they are a user, manager, or administrator, and what other users or positions report to them. Then the tasks associated with the determined position are retrieved 3208 and displayed 3210. This could include ~~task~~tasks of other positions that report to the determined position in the organizational chart. Next, the process determines 3212 whether the user can create, remove or edit a task. If so the method confirms 3214 that the position is allowed to perform the operation before performing the operation 3216 on the task and saving the updated data to the database. After either step 3212 or step 3216, the method determines 3218 whether the user is assigning a task. If so the method verifies 3220 that the position is a manger and that the position is allowed to perform the operation before assigning 3222 the task and saving the updated data to the database. After either step 3218 or step 3222, the method determines 3224 whether the user is attempting to change a position, class, project membership or some other administrative operation. If so the method verifies 3226 that the position is an administrator and that the position is allowed to perform the operation before executing the operation 3228 on the task and saving the updated data to the database. After either step 3224 or step 3228 the process is complete and ends. As can be seen from this process, the modifications to the process flow are high and seamlessly integrated into positions in the organizational chart.

Page 33, lines 13-20:

A second manner in accordance with the present invention comprises individuals obtaining information from the system 100 by electronic mail. Parts of OTIS 100 allow for email notification, wherein email messages can be formatted and sent based on the content of database 104. For example, email messages can be sent to users to inform them of work items that are scheduled to be accomplished. Further, email messages can be sent to inform managers of work-items that have not been accomplished on schedule (i.e., in a timely manner). The parameters associated with these communications are preferably stored in ~~database 104~~database 104.

Page 34, lines 2-11:

Generally, individuals with any type of right (e.g., users, managers, administrators) can view and use the screen capture feature. For example, a user can capture the error being viewed on screen by inputting a command such as ALT+PrintScreen at client 102. The user can then navigate to the input screen to save the image. For example, a "Get Image" button can be provided, which upon invocation takes the screen capture from the clipboard and through an email module (e.g., NetTransportTM) sends the image to the third module ~~where~~where it is stored in the database 104. This helps in finding the exact error. In an alternative embodiment, an individual can attach a file or multiple files to the anomaly and store it in the database 104. In an alternate embodiment, a button that brings up a help screen to guide the user through these steps can also be provided.

Page 34, line 13 through page 35, line 2:

In one aspect of the present invention, managers can view any anomaly through a novel display format. Referring to FIG. 11, an illustration is shown of an example of a user interface 1100 (review screen) for displaying to a manager for review of any anomaly. In the example shown, the review screen illustrates that the anomalies can be viewed in a tabular format of row

and columns similar to a spreadsheet. Additionally, the priority assigned to an anomaly determines what color the row is so the individual (e.g., manager) may quickly see the priority of the anomalies. The anomalies can be listed by the project 1102 they are assigned to, a user specified or unassigned 1104, bug types 1106 and input users 1108. By contrast, those users who may view anomalies are generally assigned to the particular project associated with the anomaly. For example, a user can review the pending tasks and anomalies as shown in FIG. 12. It is noted that a manager would see more details of a task and anomaly, similar to the details shown in FIG. 12, by clicking upon one of the entries in FIG. 11.

Page 36, lines 12-16:

The aspect of the present invention concerning the tracking of tasks enables supervisors to assign recurring tasks to a position class or a specific position. If an individual assigned to a particular position is unavailable, another person may step into that position and will be able to know all the tasks of that day that ~~needs~~need to be completed in an easy and efficient manner using the present invention.

Page 36, line 17-22:

There are many types of tasks allowed. There are one-time tasks that will occur once and then be done with. There are also repetitive tasks. These repetitive tasks can be broken down into flexible or fixed. A task that is assigned as flexible will be allowed to move in time (~~e.g., a holiday's weekend~~e.g., holidays, weekends, days off, etc.). A fixed task will not be allowed to move. Once a flexible or fixed indication is assigned to a task, the indication will stay assigned even if that day happens to be a holiday; and preferably, the indication should not be reassigned.

Page 43, lines 6-17:

Reference is now made to FIG. 5, which illustrates a flowchart showing an example of a process for providing status information concerning the administrative features, and for

incorporating updates thereto. At step 51, the type of administrative function is selected by the user, whereupon server 103 queries database 104 to retrieve corresponding information so that a display screen such as a web page can be formatted with the administrative status information and sent to client 102 for display 52 thereon. Control then passes to step 53, where a determination 53 is made as to whether there exists data input by an individual. If there is no data, then a default screen such as the home page is displayed 54. Otherwise, if there is data, then control passes to step 55. At step 55, a determination is made as to whether the data is formatted correctly. If not, then an error message is displayed 57. However, if the data is formatted correctly, then server 103 saves the data 56 in database 104. Thereafter, the default screen such as a home page is displayed 58, populated with the updated information.